IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

PARALLEL NETWORKS, LLC, Plaintiff, v. KOG GAMES, INC., Defendant.	Civil Action No. 13-178-RGA
PARALLEL NETWORKS, LLC, Plaintiff, v. BLIZZARD ENTERTAINMENT, INC., Defendant.	Civil Action No. 13-826-RGA
PARALLEL NETWORKS, LLC, Plaintiff, v. RELOADED GAMES, INC., Defendant.	Civil Action No. 13-827-RGA
PARALLEL NETWORKS, LLC, Plaintiff, v. SG INTERACTIVE, INC., Defendant.	Civil Action No. 13-828-RGA

MEMORANDUM OPINION

Adam W. Poff, Esq., James M. Lennon, Esq., Gregory J. Brodzik, Esq., Young Conaway Stargatt & Taylor LLP, Wilmington, DE; Brian A. Carpenter, Esq. (argued), Eric W. Buether, Esq., Christopher M. Joe, Esq., Michael D. Ricketts, Esq., Buether Joe & Carpenter, LLC, Dallas, TX, attorneys for Plaintiff Parallel Networks, LLC.

Kenneth L. Dorsney, Esq., Morris James LLP, Wilmington, DE; Eric A. Buresh, Esq. (argued), Michelle L. Marriot, Esq., Mark C. Lang, Esq., Erise IP, P.A., Overland Park, KS, attorneys for Defendants KOG Games, Inc., Reloaded Games, Inc., and SG Interactive, Inc.

Kenneth L. Dorsney, Esq., Morris James LLP, Wilmington, DE; Todd E. Landis, Esq. (argued), Eric J. Klein, Esq., Vinson & Elkins LLP, Dallas, TX, Fred I. Williams, Esq., Vinson & Elkins LLP, Austin, TX, Kellie M. Johnson, Esq., Akin Gump Strauss Hauer & Feld LLP, Dallas, TX, attorneys for Defendant Blizzard Entertainment, Inc.

August 31, 2016

Richard G. Andrews
ANDREWS, U.S. DISTRICT JUDGE:

Presently before the Court are cross-motions for summary judgment. In the cases against KOG, Reloaded, and SG Interactive ("the Reloaded Defendants"), Plaintiff Parallel Networks ("Parallel") moves for partial summary judgment of infringement with respect to the "re-allocating/re-allocate" limitation. (C.A. No. 13-178, D.I. 130).[1] The Reloaded Defendants move for summary judgment of non-infringement. (*Id.*). In the case against Blizzard, Parallel moves for partial summary judgment of infringement with respect to the "associating/associate" and "re-allocating/re-allocate" limitations. (C.A. No. 13-826, D.I. 151). Blizzard moves for summary judgment of non-infringement. (*Id.*). The issues have been fully briefed. The Court heard oral argument on May 3, 2016. (D.I. 138; C.A. No. 13-826, D.I. 161). For the reasons stated herein, Parallel's motions for partial summary judgment are both **DENIED**, the Reloaded Defendants' motion for summary judgment is **GRANTED**, and Blizzard's motion for summary judgment is **GRANTED**.

I. BACKGROUND

These are patent infringement cases. (D.I. 1). Parallel alleges infringement of U.S. Patent No. 7,188,145 ("the '145 patent"). (D.I. 130 at p. 5). On August 17, 2015, the Court held a Markman hearing. (D.I. 102). The Court construed a number of terms, including the "re-allocating/re-allocate" and "associating/associate" terms, upon which these motions focus. (D.I. 105). On October 14, 2015, pursuant to a stipulation filed by the parties, the Court allowed Parallel and the Reloaded Defendants to move for early summary judgment on the "re-allocating/re-allocate" limitation. (D.I. 117). On October 30, 2015, the Court allowed Parallel

[1] All citations will be to C.A. No. 13-178, unless otherwise indicated.

and Blizzard to move for early summary judgment on the "re-allocating/re-allocate" and "associating/associate" limitations. (C.A. No. 13-826, D.I. 140).[2]

The '145 patent is directed to a system and method for "dynamic distributed data caching." '145 patent Cover Page. The claims recite a "cache community" with at least "one peer." '145 patent Abstract.[3] "Each peer has an associated first content portion . . . indicating content to be cached by the respective peer." *Id.* When a "client" joins this cache community, the cache community's "peer list," which "indicates the peers in the cache community," "is updated to include the client." *Id.* Then, "[a] respective second content portion . . . is associated with each peer based on the addition of the client." *Id.* "The second content portion is distinct from the first content portion." *Id.* at 1:52-53. The "invention . . . reallocates the data to be cached by particular members of the distributed cache community based on the addition and subtraction of members to the distributed cache community." *Id.* at 4:32-34. The claimed invention allows for "decreased utilization of expensive connections to the Internet and increased utilization of cheaper local area network connections and broadband connections." *Id.* at 4:10-12. "By caching content at local machines on a local area network or on broadband connections to an Internet Service Provider, response time to requests for content is decreased by retrieving the content from local machines." *Id.* at 4:14-18.

The '145 patent has two independent claims: claim 1, a system claim, and claim 15, a method claim. The "re-allocating/re-allocate" and "associating/associate" limitations appear in both independent claims. During claim construction, the Court construed "re-allocating/re-

[2] Parallel has not raised a doctrine of equivalents argument in either case. Thus, both sets of cross-motions relate only to literal infringement.

[3] The parties do not dispute the meaning of "cache" or "caching." "Generally, it means storing data in a way that is relatively cheap to access (for example, in terms of time or bandwidth) in order to avoid getting it from a place that is more expensive to access." (D.I. 130 at 26).

allocate the cache storage of the content among the peers in the cache community" to mean "re-allocating/re-allocate the portions of the content to be stored by each peer's cache storage." (D.I. 105 at p. 2). The Court construed "associating/associate" to mean "[indicating/indicate] content to be cached by the client." (*Id.* at p. 3).

Claim 1, which is illustrative, reads:

> A method for dynamic distributed data caching comprising:
>
> providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with content obtained from a second side of the point of presence, the content being cached by the at least one peer;
>
> allowing a client to join the cache community;
>
> updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community;
>
> associating the content with the client based on joinder of the client;
>
> re-allocating the cache storage of the content among the peers in the cache community in response to allowing the client to join the community.

'145 patent at 28:49-65.[4]

There are two accused products in these cases: Pando and the Blizzard Downloader. The two accused products are similar, but not identical. The relevant facts surrounding each are described separately.

A. Pando

The Reloaded Defendants and Parallel submitted a joint stipulation of facts. (D.I. 130 at pp. 1-4). I will summarize the relevant facts.[5] The Reloaded Defendants distributed video games through a system called Pando. Pando delivered content to end users through multiple

[4] While claim 1 recites "associating" and "re-allocating," claim 15 recites "associate" and "re-allocate."

[5] All facts are gleaned from the joint stipulation, unless otherwise indicated.

avenues, including direct downloads from a Content Delivery Network ("CDN") and peer-to-peer downloads from other users. These cases focus on the peer-to-peer aspect of Pando.

Game packages for video games are very large, often several gigabytes in size. (D.I. 131-2 at 80). To effectively share such large files, Pando allowed users to transfer small, manageable portions of data amongst themselves. The relevant unit of data is called is called a "piece." Pando divided game packages into "pieces," which were typically one megabyte in size.

When a user sought to download a game using Pando, that user downloaded the Pando Game Downloader, which in turn downloaded and installed the Pando Media Booster (the "PMB client"). The PMB client then connected to a server, called a tracker server, which kept a list of all the other active PMB clients downloading the same game package. This collection of clients, all downloading the same game, was called the "swarm." Before it began downloading the desired game package, a given PMB client ("the subject PMB client") first, and at a regular interval thereafter, sent an "Announce Message" to the tracker server. After the tracker server received the Announce Message, it added the subject PMB client to the list of PMB clients in the swarm.[6]

In response to the subject PMB client sending an Announce Message to the tracker server, the tracker server sent the subject PMB client a list of some subset of the clients in the swarm ("the Selected List"). The tracker server chose PMB clients for the Selected List based on network proximity and random selection. In response to each Announce Message, the tracker

[6] Every PMB client in the swarm was deemed either a "seed" or a "leech." A seed had successfully downloaded 100% of the game package, while a "leech" was in the process of downloading the game package. The tracker server knew whether a PMB client was a leech or a seed and how much data a given PMB client had downloaded, but did not know which pieces that PMB client had downloaded.

server sent the subject PMB client a new Selected List, which the subject PMB client merged with previously received Selected Lists to create a "Merged List." As a result of this process, two PMB clients in the swarm likely had non-identical Merged Lists.

The subject PMB client attempted to form connections with some or all of the PMB clients in its Merged List. A successful connection was referred to as an "established connection." A PMB client had a maximum limit on the number of established connections it could have at one time.

After establishing a connection, the subject PMB client and the PMB client to which it connected exchanged "Bit Field Messages." A Bit Field Message contained an inventory, called a bit field, for each piece of the game package a PMB client had. In other words, a Bit Field Message indicated which pieces a given PMB client possessed and which it did not. Whenever a PMB client successfully downloaded a new piece, it sent a "Have Message" to each of its established connections, thereby informing its established connections that it had a new piece of data to share.

Whenever the subject PMB client received a Bit Field Message or a Have Message from an established connection, it independently determined what, if anything, it should request from that established connection. The parties dispute the ways in which a PMB client selected which piece to download. Of central importance to Parallel's infringement theory is the "Rarest-First Algorithm." Parallel contends that the Rarest-First Algorithm was "the only strategy used by the PMB [c]lient to generate requests for content." (D.I. 130 at 15). The Reloaded Defendants argue that the algorithm was "not necessarily used to make requests, and in many instances will never be used at all," as a PMB client "must run at least three different sub-algorithms that request portions of content" before it may "get[] to the Rarest-First Algorithm." (*Id.* at 51). In addressing the Reloaded Defendants' motion, I accept that the Rarest-First Algorithm was used by Pando.[7]

The parties do not dispute how the Rarest-First Algorithm functioned. When using the Rarest-First Algorithm, the subject PMB client generated a list of pieces which a particular established connection had, and which it had not requested from any other PMB client. (D.I. 130 at 16). The subject PMB client then assessed which piece, within that list, was the rarest.[8] (*Id.*). It then sought to download that rarest piece, or if there were several equally-rare pieces, it downloaded one of the equally-rare pieces. (*Id.*).

The downloading process continued until a PMB client obtained 100% of the game package. That is, the goal of every user was to obtain 100% of the game package, since the game could not be played until the entire game package was downloaded.

B. Blizzard Downloader

Blizzard and Parallel did not submit a joint stipulation of facts. The parties do, however, agree on most of the material facts.[9]

To deliver game content to users, Blizzard used two software applications: the Blizzard Installer and the Blizzard Downloader ("the BDL client"). (C.A. No. 13-826, D.I. 151 at 26, 50). This case concerns the BDL client. Blizzard's process began with game packages being divided

[7] Since Parallel is the non-moving party in the Reloaded Defendants' dispositive motion, I will "assume the truth of the [Parallel's] evidence, and draw all justifiable inferences in [its] favor." *Clark v. Sears, Roebuck & Co.*, 827 F. Supp. 1216, 1218 (E.D. Pa. 1993) (citing *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 255 (1986)).
[8] The process for determining rarity is as follows. Each PMB client maintained an integer value associated with each piece, called the "Have Count," which indicated the number of its established connections that had received that piece. (*Id.*). Upon receiving a Bit Field Message or a Have Message from one of its established connections, the subject PMB client updated the Have Count associated with a particular piece. (*Id.*).
[9] Blizzard relies on a declaration from fact witness Jay Lauffer, a senior software engineer at Blizzard. (C.A. No. 13-826, D.I. 153-10 at 111). Parallel relies on declaration from its expert, Dr. Peter Reiher. (C.A. No. 13-826, D.I. 153-9 at 8).

into pieces, which were typically 256 kilobytes in size. (*Id.* at 35, 50). Each piece was "hashed," thereby giving each piece a unique identifier, so that each piece could be verified as authentic. (*Id.*). After all pieces were hashed, all of the hashes for the game package were hashed again into a hash called "info_hash." (*Id.*). The info_hash, along with information about the hash for each piece, was stored in a "torrent" file. (*Id.* at 35-36, 50). This torrent file also contained a list of the files to be downloaded, and a list of the direct download addresses which allowed a user to download pieces from CDNs. (*Id.*).

When a user sought to download a game from Blizzard, the torrent file associated with that game was loaded into the user's BDL client ("the subject BDL client"). (*Id.* at 36, 51). After the user received the torrent file, the user began downloading game content through the CDNs provided in the torrent file. (*Id.*). If a user had enabled peer-to-peer sharing, the subject BDL client then extracted the info_hash value from the torrent file and sent it to Blizzard's tracker server through an Announce Message. (*Id.*). The tracker server then used a method called "Torrent::FindTracker" to retrieve a "Tracker" object, based on the info_hash value sent in the Announce Message. (*Id.* at 52).[10] This Tracker object was copied to a data field called "m_tracker" within the "Peer" object that the tracker server had associated with the joining client. (*Id.*).

Thereafter, the tracker server sent an "Announce Response" to the subject BDL client. That response included a Selected List, a list of BDL clients which were some subset of the total BDL clients in the swarm. (*Id.*). The subject BDL client then attempted to connect to some or all of the BDL clients in its Selected List. (*Id.* at 13). As with Pando, successful connections

[10] With respect to this process, Blizzard does not address some of the facts described by Parallel. I therefore understand them to be undisputed. To the extent there is a dispute, I will assume the truth of Parallel's evidence.

were referred to as established connections. (*Id.* at 14). Established connections then exchanged Bit Field Messages, each of which detailed which pieces of the game package a given BDL client had. (*Id.*). After the subject BDL client received a Bit Field Message, it selected which piece to download, if any, from that established connection. (*Id.*). After successfully downloading a piece from another BDL client in the swarm, a BDL client sent a "Have Message" to all of its established connections, informing them that it had a new piece available for download. (*Id.*).

Blizzard created a priority system for its game packages.[11] (*Id.* at 37, 54-55). Each piece of the game package was placed into one of three buckets: a red bucket, a yellow bucket, and a green bucket. (*Id.* at 37). The red bucket included all game files required to begin playing the game. (*Id.*). Users had to download all of the files within the red bucket before beginning to download files in the yellow bucket, and had to download all of the files in the yellow bucket before beginning to download the files in the green bucket. (*Id.*). While downloading files within a particular bucket, the BDL client used a "Rarest-First Algorithm." (*Id.* at 16, 54-55). The Rarest-First Algorithm employed by the BDL client is identical, in all material respects, to that employed by Pando. (*Id.* at 16-17).

If a user did not exit the swarm, that user would eventually download 100% of the game package. (*Id.* at 38, 43). After a user finished downloading the game, that user was considered a "seed," and soon exited the swarm. (*Id.*).

[11] There are three versions of the BDL client. The first two versions are accused, but the third is not. This priority system was only implemented in the second version of the BDL client. Blizzard previously agreed "that it [would] not present any non-infringement arguments . . . on the 're-allocating' and 'associating' limitations that peer-to-peer functionality differ[ed] among different versions of Blizzard source code." (C.A. No. 13-826, D.I. 140 at p. 2). Thus, Blizzard is precluded from making any non-infringement arguments based on the priority system, or based on the operation of the third version of the BDL client.

II. LEGAL STANDARD

"The court shall grant summary judgment if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a). The moving party has the initial burden of proving the absence of a genuinely disputed material fact relative to the claims in question. *Celotex Corp. v. Catrett*, 477 U.S. 317, 330 (1986). Material facts are those "that could affect the outcome" of the proceeding, and "a dispute about a material fact is 'genuine' if the evidence is sufficient to permit a reasonable jury to return a verdict for the nonmoving party." *Lamont v. New Jersey*, 637 F.3d 177, 181 (3d Cir. 2011) (quoting *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986)). The burden on the moving party may be discharged by pointing out to the district court that there is an absence of evidence supporting the non-moving party's case. *Celotex*, 477 U.S. at 323.

The burden then shifts to the non-movant to demonstrate the existence of a genuine issue for trial. *Matsushita Elec. Indus. Co. v. Zenith Radio Corp.*, 475 U.S. 574, 586-87 (1986); *Williams v. Borough of West Chester, Pa.*, 891 F.2d 458, 460-61 (3d Cir. 1989). A non-moving party asserting that a fact is genuinely disputed must support such an assertion by: "(A) citing to particular parts of materials in the record, including depositions, documents, electronically stored information, affidavits or declarations, stipulations . . . , admissions, interrogatory answers, or other materials; or (B) showing that the materials cited [by the opposing party] do not establish the absence . . . of a genuine dispute . . . ." Fed. R. Civ. P. 56(c)(1).

When determining whether a genuine issue of material fact exists, the court must view the evidence in the light most favorable to the non-moving party and draw all reasonable inferences in that party's favor. *Scott v. Harris*, 550 U.S. 372, 380 (2007); *Wishkin v. Potter*, 476 F.3d 180, 184 (3d Cir. 2007). If the non-moving party fails to make a sufficient showing on an essential element of its case with respect to which it has the burden of proof, the moving party is entitled to judgment as a matter of law. *See Celotex*, 477 U.S. at 322.

A patent is infringed when a person "without authority makes, uses, offers to sell, or sells any patented invention, within the United States . . . during the term of the patent." 35 U.S.C. § 271(a). A two-step analysis is employed in making an infringement determination. *See Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 976 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996). First, the court must construe the asserted claims to ascertain their meaning and scope. *See id.* The trier of fact must then compare the properly construed claims with the accused infringing product. *See id.* at 976. This second step is a question of fact. *See Bai v. L & L Wings, Inc.*, 160 F.3d 1350, 1353 (Fed. Cir. 1998).

"Literal infringement of a claim exists when every limitation recited in the claim is found in the accused device." *Kahn v. Gen. Motors Corp.*, 135 F.3d 1472, 1477 (Fed. Cir. 1998). "If any claim limitation is absent from the accused device, there is no literal infringement as a matter of law." *Bayer AG v. Elan Pharm. Research Corp.*, 212 F.3d 1241, 1247 (Fed. Cir. 2000). If an accused product does not infringe an independent claim, it also does not infringe any claim depending thereon. *See Wahpeton Canvas Co. v. Frontier, Inc.*, 870 F.2d 1546, 1553 (Fed. Cir. 1989). The patent owner has the burden of proving infringement and must meet its burden by a preponderance of the evidence. *See SmithKline Diagnostics, Inc. v. Helena Lab. Corp.*, 859 F.2d 878, 889 (Fed. Cir. 1988) (citations omitted).

When an accused infringer moves for summary judgment of non-infringement, such relief may be granted only if at least one limitation of the claim in question does not read on an element of the accused product. *See Chimie v. PPG Indus., Inc.*, 402 F.3d 1371, 1376 (Fed. Cir. 2005); *see also TechSearch, L.L.C. v. Intel Corp.*, 286 F.3d 1360, 1369 (Fed. Cir. 2002)

("Summary judgment of noninfringement is . . . appropriate where the patent owner's proof is deficient in meeting an essential part of the legal standard for infringement, because such failure will render all other facts immaterial."). Thus, summary judgment of non-infringement can only be granted if, after viewing the facts in the light most favorable to the non-movant, there is no genuine issue as to whether the accused product is covered by the claims (as construed by the court). *See Pitney Bowes, Inc. v. Hewlett–Packard Co.*, 182 F.3d 1298, 1304 (Fed. Cir. 1999).

III. ANALYSIS

A. The Reloaded Defendants' Motion for Summary Judgment

Parallel's infringement theory centers on the execution of the Rarest-First Algorithm under certain circumstances. Parallel contends that the "re-allocating/re-allocate" limitation is met in two ways. First, when an existing peer[12] adds to its Have Count in response to receiving the first Have Message from a new peer that has just joined the swarm, [13] its view of rarity will be altered. After executing the Rarest-First Algorithm, the existing peer may request a different piece than it otherwise would have, had the new peer not joined the swarm and sent its first Have Message. Specifically, Parallel argues that the reallocation limitation is met when an existing peer adds to its Have Count and executes the Rarest-First Algorithm. Second, when a peer joins the swarm and sends its first Have Message to its established connections, the established connections will alter their respective Have Counts. For each peer, after executing the Rarest-First Algorithm, "the order of the [p]ieces to be downloaded and stored in the future" may be affected. (D.I. 130 at 14). Specifically, Parallel argues that the reallocation limitation is met

[12] "Peer" is the term used by the '145 patent. In the accused Pando system, each peer is a PMB client. An existing peer is a PMB client that has already joined the swarm.
[13] The claims require that the reallocation occur "in response to allowing the client to join the community." '145 patent at 28:63-65, 30:1-4. Thus, Parallel contends that this limitation is satisfied when a Have Message is sent from a new peer that was allowed to join the swarm.

when the peer that has just joined the swarm sends its first Have Message. These arguments make up two sides of the same coin. According to Parallel, sending or receiving a Have Message constitutes a reallocation, because, following the execution of the Rarest-First Algorithm, the Have Message may alter the order in which pieces are downloaded.

The reallocation limitation requires that there be a "re-allocation" of "the portions of the content to be stored by each peer's cache storage." (D.I. 105 at p. 2). The use of "portions" in the construction is important. At the Markman hearing, I noted that the "primary dispute" between the parties was "whether each peer [could] cache 100 percent of the content." (D.I. 102 at 115). I concluded that each peer could not, and stated that "each peer receives some but not all of the content." (*Id.*). "[I]f each peer had all the data, the invention would face the same problems that existed in the prior art." (*Id.*).

Parallel's infringement theory is fatally flawed. In the accused Pando system, the "portions of the content to be stored" by each PMB client are never altered. In fact, every user will use the same amount of space to store the same data. In other words, each PMB client will download every piece of the game package, regardless of the order in which those pieces are downloaded. Accordingly, the portions of the content to be stored are never reallocated, or, indeed, altered in any way.

Parallel urges the Court to look at the transfer of a particular piece, from peer to peer, in isolation. When a new peer joins the swarm and sends its first Have Message, Parallel argues, an existing peer's view of rarity is affected. Then, upon execution of the Rarest-First Algorithm, a PMB client will download pieces in an order that depends on its view of rarity. When the reordering occurs as a result of a Have Message being sent or received, Parallel argues that a reallocation has occurred. According to Parallel, this "re-allocation" occurs every time a new peer sends a Have Message, up until an existing peer actually has 100% of the content.[14] No reasonable juror could reach that conclusion.

When the '145 patent recites reallocation, it envisions some degree of divided responsibility for portions of content. For instance, the specification explains that the "[d]ynamic cache application 428 . . . reallocates the content 460 to be cached by particular members 412 and master 410 so that the newly added client 404 is responsible for some subset of content 460 cached in community 402." '145 patent at 21:10-13; *see also id.* at 4:32-34 ("The present invention also reallocates the data to be cached by particular members of the distributed cache community based on the addition and subtraction of members to the distributed cache community."). Whether or not there is some overlap in that divided responsibility[15]—i.e., some level of data duplication between peers in the cache community—does not alter the necessary conclusion that "each peer receives some [portion,] but not all of the content." (D.I. 102 at 115). The claimed invention requires that a new peer's entry into the swarm causes the particular portions of content that each peer is responsible for storing to change.

With Pando, the portions of the content to be stored by each peer never changes. Each PMB client constantly progresses toward its goal of receiving 100% of the content. When a

[14] In light of the Court's claim construction ruling, Parallel argues that infringement occurs during "the period of time between a [p]eer being allowed to join and the [p]eer obtaining 100% of the [g]ame [p]ackage." (D.I. 130 at 55).

[15] The parties dispute whether a reallocation inherently requires some removal of content, rather than duplication. In other words does reallocation require moving a piece from peer A to peer B, or can it simply be copying that piece, such that peer A and peer B both have a copy? I think the '145 patent contemplates both. As the specification explains:

> Redistribution may comprise actually copying cached content from one client . . . to another, or removing or expiring content no longer cached at a particular server. Simply removing the cached items may cause a request that could have been satisfied by the community cache to be forwarded to the origin server, but avoids the bandwidth usage associated with copying. The decision whether to copy or remove may be configured at modules 124, 144 and 164 and may consider bandwidth usage issues . . . .

'145 patent at 16:41-50. Although the parties contest this issue at length, I do not think it is material to the outcome here.

PMB client adds to its Have Count and executes the Rarest-First Algorithm, all that is decided is which piece the PMB client should copy from another PMB client at that instant. The PMB client, from the moment it joined the swarm, always sought to download that piece. In other words, the PMB client would have downloaded that piece regardless of whether a new PMB client ever joined the swarm and sent a Have Message. This cannot constitute a reallocation of "the portions of the content to be stored by each" PMB client. (D.I. 105 at p. 2).

Parallel notes that "the act of incrementing the Have Count affecting [an] [e]xisting [p]eer's view of rarity . . . constitute[s] a re-allocation of the order the [p]ieces will be downloaded in the future." (D.I. 130 at 29). Herein lies the problem with Parallel's theory. The '145 patent does not require a reallocation "of the order the [p]ieces will be downloaded;" it requires a reallocation of the "portions of the content to be stored" among the peers. Additionally, Parallel's expert is correct that "'to be' is prospective in nature." (D.I. 131-4 at 78). Thus, a reallocation would require a change in what each peer ultimately seeks to store. That does not happen in the Pando system.

Elsewhere in its briefing, Parallel argues that "[t]he 'Have Count' . . . affected each receiving [p]eer's choice . . . as to what [p]ieces of content . . . it would request and store in the future." (D.I. 130 at 30). This statement not only contradicts Parallel's own arguments about "re-allocat[ing] . . . the order the [p]ieces will be downloaded," but also the stipulated facts. (*Id.* at 29). In the stipulated facts, Parallel agreed that "[t]he goal of each PMB [c]lient that participated in a [s]warm was to successfully download and store 100% of the [g]ame [p]ackage." (*Id.* at 10). Thus, the Have Count does not affect the pieces of content that would be requested and stored by a peer in the future. It only affects the order in which those pieces will be requested. To put it succinctly, a reordering is not a reallocation. Parallel's infringement theory is premised on an assumption that changing the order in which a particular client downloads content from another constitutes a reallocation of the portions of content among the peers in the cache community. No reasonable juror could so conclude.

Parallel has failed to show a genuine dispute of material fact which would preclude the entry of summary judgment.

B. Parallel's Motion for Partial Summary Judgment Against the Reloaded Defendants

Parallel moves for partial summary judgment of infringement on the "re-allocating/re-allocate" limitation. Having concluded that the Reloaded Defendants are entitled to summary judgment of non-infringement, Parallel's motion for summary judgment is denied.

C. Blizzard's Motion for Summary Judgment

Blizzard moves for summary judgment of non-infringement. While the Reloaded Defendants focused solely on the "re-allocating/re-allocate" limitation, Blizzard focuses on both that limitation and the "associating/associate" limitation. Since I find the "re-allocating/re-allocate" limitation to be dispositive, I need not and do not address the "associating/associate" limitation.

Parallel's infringement theory for the BDL client is identical, in all material respects, to its infringement theory for Pando. Generally speaking, Parallel argues that sending or receiving a Have Message constitutes a reallocation, because, following the execution of the Rarest-First Algorithm, the Have Message may alter the order in which pieces are downloaded.

Parallel's infringement theory fails with respect to Blizzard for the same reasons it failed with respect to the Reloaded Defendants. As with Pando, every BDL client downloads 100% of the game package. That is, every user utilizes the same amount of space to store the exact same data. There is no divided responsibility, as no user ultimately stores less than 100% of the content. While the Rarest-First Algorithm may alter the order in which certain pieces are downloaded, it does not alter the portions of the content to be stored by each user in the swarm.

Therefore, a reasonable juror could not conclude that the accused BDL client satisfies the "re-allocating/re-allocate" limitation. Blizzard's motion for summary judgment of non-infringement is granted.

D. Parallel's Motion for Partial Summary Judgment Against Blizzard

Parallel moves for partial summary judgment of infringement on the "re-allocating/re-allocate" and the "associating/associate" limitations. Having concluded that Blizzard is entitled to summary judgment of non-infringement, Parallel's motion for partial summary judgment is denied.

IV. CONCLUSION

For the reasons set forth above, the Reloaded Defendants' motion for summary judgment of non-infringement is **GRANTED**, Blizzard's motion for summary judgment of non-infringement is **GRANTED**, and Parallel's motions for partial summary judgment of infringement are **DENIED**. An appropriate order will be entered.